Knowledge Redundancy Approach to Reduce Size in Association Rules

Julio César Díaz Vera
University of Informatics Sciences, Havana, Cuba
E-mail: jcdiaz@uci.cu

Guillermo Manuel Negrín Ortiz
University of Informatics Sciences, Havana, Cuba
E-mail: gmnegrin@uci.cu

Carlos Molina
University of Jaen, Jaen, Spain
E-mail: carlosmo@ujaen.es

Maria Amparo Vila
University of Granada, Granada, Spain
E-mail: vila@decsai.ugr.es

Keywords: association rule mining, redundant rules, knowledge guided post-processing

Received: June 19, 2019

Association Rules Mining is one of the most studied and widely applied fields in Data Mining. However, the discovered models usually result in a very large set of rules; so the analysis capability, from the user point of view, is diminishing. Hence, it is difficult to use the found model in order to assist in the decision-making process. The previous handicap is heightened in the presence of redundant rules in the final set. In this work, a new definition of redundancy in association rules is proposed, based on user prior knowledge. A post-processing method is developed to eliminate this kind of redundancy, using association rules known by the user. Our proposal allows finding more compact models of association rules to ease its use in the decision-making process. The developed experiments have shown reduction levels that exceed 90 percent of all generated rules, using prior knowledge always below ten percent. So, our method improves the efficiency of association rules mining and the exploitation of discovered association rules.

Povzetek: Opisan je sistem za zmanjševanje števila in dolžine pravil s pomočjo analize redundantnosti za metode asociativnega učenja.

1 Introduction

Mining for association rules has been one of the most studied fields in data mining. Its main goal is to find unknown relations among items in a database.

Given a set of items \( I \) which contains all the items in the domain and a transactional database \( D \) where every transaction is composed by a transaction id (\( tid \)) and a set of items, subset of \( I \) (itemset).

An association rule is presented as an implication \( X \rightarrow Y \) where \( X \) is the antecedent and \( Y \) is the consequent of the rule. Both \( X \) and \( Y \) are itemsets and usually, but not necessarily, they check \( X \cap Y = \emptyset \) property. Association rules reflect how much the presence of the rule antecedent influences the presence of the rule consequent in the database records.

What generally makes a rule meaningful are two statistical factors: support and confidence. The support of a rule \( \text{supp}(X \rightarrow Y) \) refers to the portion of the database transaction for which \( X \cap Y \) is true while confidence \( \text{conf}(X \rightarrow Y) \) is a measure of certainty to evaluate the validity of the rule, it is a measure for the portion of record which contains \( Y \) from those that contain \( X \). The problem with association rule mining deals with finding all the rules that satisfy a user-given threshold for support and confidence. Most algorithms face the challenge in a two steps procedure

1. Find all the itemsets which support value is equal or greater than the support threshold.
2. Generate all association rules \( X \rightarrow (Y - X) \), considering: \( Y \) is a frequent itemset, \( X \subseteq Y \), and \( \text{conf}(X \rightarrow Y) \) is equal or greater than the confidence threshold value.

The discovering of meaningful association rules can help in the decision-making process but the quite large number of rules usually makes it difficult for decision-makers in order to process, interpret and apply them. A significant part of the rules presented to the user are irrelevant because they are obvious, too general, too specific or because they are not relevant for the decision topic. Several methods were proposed in the literature to overcome this handicap such
as interest measures development, concise representations of frequent itemsets and redundancy reduction. Section 2 discusses some of the most important works in the field.

This paper proposes a new approach to deal with redundancy, taking into account user previous knowledge about the studied domain. Previous knowledge is used to detect and prune redundant rules. We adapt the concept of redundancy and we propose a procedure to develop the redundancy reduction process in the post-processing stage.

The paper is organized as follows. Section 2 discusses related work. In section 3 we propose an algorithm to find and prune redundant rules. In section 4 the proposed algorithm is used over three datasets one with data about financial investment [1], other with data about the USA census [2] and the other with data about Mushrooms [2]. Section 5 closes the paper with conclusions.

2 Related work

Interestingness is difficult to define quantitatively [3] but most interestingness measures are classified in objective measures and subjective measures. Objective measures are domain-independent, one of them is the interestingness which is expressed in terms of statistic or information theory applied over the database. Several surveys [4, 5, 6] summarize and compare objective measures. The explosion of objective measures has raised a new problem: What are the best metrics to use in a specific situation and a particular application field? Several papers attempt to solve it [8, 9] but it is far from being solved. The correlation between 11 objective rule interestingness measures and real human interest over eight different datasets were computed in [10] and there was not a clear “winner”, the correlation values associated with each measure varied considerably across the eight datasets.

Subjective measures were proposed in order to involve explicitly user knowledge in the selection of interesting rules so that the user can make a better selection. According to [11] subjective measures are classified in:

- Unexpectedness: a pattern is interesting if it is surprising to the user.
- Actionability: a pattern is interesting if it can help the user to take some actions.
- Precise Knowledge. Item taxonomies concept was developed in [18] and [19]. They developed a specification language to express expectations and goals. Three levels of specification were established: General Impressions, Reasonably Precise Concept and Precise Knowledge. Item taxonomies concept was integrated in the specification languages in order to generalize rule selection. The matching process involved a syntactic comparison between antecedent/consequent elements. Thus, each element in the general impression should find a correspondent in the association rule.

This approach is still in a development stage, no further advances were published, so it is not functional.
- General Impressions: were presented in [17] and later developed in [18] and [19]. They developed a specification language to express expectations and goals. Three levels of specification were established: General Impressions, Reasonably Precise Concept and Precise Knowledge. Item taxonomies concept was integrated in the specification languages in order to generalize rule selection. The matching process involved a syntactic comparison between antecedent/consequent elements. Thus, each element in the general impression should find a correspondent in the association rule.
- Logical Contradiction: was developed in [20]. It consists in extracting only those patterns which logically contradict the consequence of the corresponding belief.
An association rule $X \rightarrow Y$ is unexpected with respect to some belief $A \rightarrow B$ if:

- $Y \land B \models \texttt{FALSE}$ and $Y$ and $B$ are in logical contradiction;
- $X \land B$ has an important support in the database. This condition eliminates those rules which could be considered unexpected, but not those concerning the same transaction in the database;
- $A, X \rightarrow B$ exists.

- Preference Model: was proposed in [21]. It is a specific type of user knowledge representing how the basic knowledge of the user, called knowledge rules ($\mathcal{K}$), will be applied over a given scenario or tuples of the database. The user proposes a covering knowledge ($\mathcal{Ct}$) for each tuple ($t$) - a subset of the knowledge rule set $K$ that the user prefers to apply to the tuple $t$. The approach validates the transactions which satisfy the extracted rule.

All the previously presented works use some kind of knowledge to reduce the number of useless association rules in the final set. In this way, our approach is similar to them but there are some remarkable differences.

Like in templates our approach uses the syntactical notation of association rules to represent knowledge. Templates use this knowledge to constraint the structure of selected rules, pruning out those rules which do not satisfy the template but produce a lot of association rules with similar information. On the other hand, we use the knowledge to remove those rules with similar information, presenting to the user a set of unexpected rules that can help him to better understand the underlying domain.

The approach followed by Belief tries to find just unknown rules, this is our main goal too but, they use a complex and fixed formal knowledge representation based on first order logic and degrees of belief with no clear way of building and maintaining the belief system. Instead, we use a simpler and natural rule-based form of knowledge, focused on the enhanced capability to increase interactively the knowledge system.

### 2.1 Rule redundancy reduction

Research community accepts the seminal definition of association rule redundancy given in [22] “an association rule is redundant if it conveys the same information - or less general information - than the information conveyed by another rule of the same usefulness and the same relevance”. But several formal definitions have been proposed over time. In table 1, a sample transactional database is presented. Defining a support threshold of 0.15 and a confidence threshold of 0.75, an association rule model with 92 rules is obtained. It is used to show redundancy definitions.

<table>
<thead>
<tr>
<th>Income</th>
<th>Balance</th>
<th>Sex</th>
<th>Unemployed</th>
<th>Loan</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>High</td>
<td>F</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>M</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>M</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>F</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>M</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>F</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>M</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>F</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>M</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>F</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>M</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1: Sample transactions

**Definition 1.** Minimal non-redundant association rules[22]: An association rule $R : X \rightarrow Y$ is a minimal non-redundant association rule if there is not an association rule $R_1 : X_1 \rightarrow Y_1$ with:

- $\text{support}(R) \leq \text{support}(R_1)$
- $\text{confidence}(R) \leq \text{confidence}(R_1)$
- $X_1 \subseteq X$ and $Y \subseteq Y_1$

From data on table 1 we can obtain the rules:

$R : \{\text{[balance],[medium]}\} \rightarrow \{\text{[income],[low],[loan],[no]}\}$ $\text{supp} = 0.25$, $\text{conf} = 0.75$ and $R_1 : \{\text{[balance],[medium]}\} \rightarrow \{\text{[loan],[no]}\}$ $\text{supp} = 0.25$, $\text{conf} = 0.75$. According to definition 1 $R$ is a redundant rule. No new information is provided by its inclusion into the association rules model.

Several works have been developed to prune that kind of redundancy. Mining Closed Associations, uses frequent closed itemsets [23] tries to produce the set of minimal generators for each itemset. The number of closed association rules is linear to the number of closed frequent itemsets. It can be large for sparse and large datasets.

The Generic Basis (GB) and the Informative Basis (IB) [22] used the Galois connections to propose two condensed basics that represent non-redundant rules. The Gen-GB and Gen-RI algorithms were presented to obtain a generic basis and a transitive reduction of the IB. The reduction ratio of IB was improved by [24] maximal closed itemsets. The Informative Generic Basis [25] also uses the Galois connection semantics but taking the support of all frequent itemsets as an entry, so it can calculate the support and confidence of derived rules. The augmented Iceberg Galois lattice was used to construct the Minimal Generic Basis (MGB) [26]. The concept of generator was incorporated into high utility itemsets mining in [27].

The redundancy definition presented in definition 1 requires that a redundant rule and its corresponding non-redundant rule must have identical confidence and identical support. From data on table 1 we can obtain the...
rules:
\[
R : \{\text{[income],[high]}, \text{[unemployed],[no]}\} \rightarrow \{\text{[loan],[yes]}\} \quad \text{supp} = 0.33, \text{conf} = 1.0, \text{and}
R_1 : \{\text{[income],[high]}\} \rightarrow \{\text{[loan],[yes]}\} \quad \text{supp} = 0.41, \text{conf} = 1.0 \]

those rules are non-redundant ones, but the consequent of \( R \) can be obtained from \( R_1 \) a rule with the same confidence and fewer conditions. So without \( R \) the same results are achieved, rule \( R \) must be a redundant rule. Xu [28] formalizes this kind of redundancy in definition 2.

**Definition 2. Redundant rules[28]:** Let \( X \rightarrow Y \) and \( X_1 \rightarrow Y_1 \) be two association rules with confidence \( cf \) and \( c_{f1} \), respectively. \( X \rightarrow Y \) is said to be a redundant rule to \( X_1 \rightarrow Y_1 \) if

\[
- X_1 \subseteq X \quad \text{and} \quad Y \subseteq Y_1
\]

\[
- cf \leq c_{f1}
\]

Based on definition 2 the Reliable basis was proposed. It consists of two bases the ReliableApprox used in partial rules, and ReliableExact used in exact rules. Frequent closed itemsets are used to perform the reliable redundancy reduction process. It generates rules with minimal antecedent and maximal consequent. The reliable basis removes a great amount of redundancy without reducing the inference capacity of the remaining rules. Phan [29] uses a more radical approach to define redundancy see definition 3.

**Definition 3. Representative association rules[29]:** Let \( X \rightarrow Y \) an association rule. \( X \rightarrow Y \) is said to be a representative association rule if there is not other interesting rule \( X_1 \rightarrow Y_1 \) such that \( X_1 \subseteq X \) and \( Y \subseteq Y_1 \).

The redundancy definitions presented above do not guarantee the exclusion of all non-interesting patterns of the final model. Example 1 shows a group of rules with no new information to the user, and they are not classified as redundant by the previous definitions.

**Example 1. A set of redundant rules from data in table 1**
Let’s see a subset of association rules obtained from table 1:

\[
R_1 : \{\text{[income],[high]}\} \rightarrow \{\text{[loan],[yes]}\}
R_2 : \{\text{[sex],[female]}, \text{[unemployed],[no]}\} \rightarrow \{\text{[income],[high]}\}
R_3 : \{\text{[sex],[female]}, \text{[unemployed],[no]}\} \rightarrow \{\text{[income],[high]}, \text{[loan],[yes]}\}
R_4 : \{\text{[sex],[female]}, \text{[unemployed],[no]}\} \rightarrow \{\text{[loan],[yes]}\}
R_5 : \{\text{[income],[high]}, \text{[loan],[yes]}\} \rightarrow \{\text{[unemployed],[no]}\}
R_6 : \{\text{[income],[high]}, \text{[loan],[yes]}, \text{[sex],[male]}\} \rightarrow \{\text{[unemployed],[no]}\}
R_7 : \{\text{[balance],[high]}, \text{[income],[high]}, \text{[loan],[yes]}\} \rightarrow \{\text{[unemployed],[no]}\}
\]

If we analyze the rules \( R_1 \) and \( R_3 \) we see that item \{loan,[yes]\} in \( R_3 \) consequent provides no new information, because this is known by \( R_1 \). So rule \( R_3 \) is redundant but this kind of redundancy is not detected by the previous definitions. Analyzing rules \( R_1, R_2 \) and \( R_4 \) we can check that combining, transitively, of \( R_1 \) and \( R_2 \) it will produce \( R_4 \) so, \( R_4 \) is redundant. One more time this kind of redundancy is not detected by previous definitions. In \( R_5, R_6 \) and \( R_7 \) antecedent the item \{loan,[yes]\} provides no new information because this is known by \( R_1 \). It is redundant and must be pruned, but it can not be detected by redundancy definitions.

## 2.2 Post-processing
Since the year 2000, the interest in post-processing methods in association rules has been increasing. Perhaps the most accurate definition of post-processing tasks was done by Baesens et al. [30] Post-processing consists of different techniques that can be used independently or together: pruning, summarizing, grouping and visualization. We have a special interest in pruning techniques that prune those rules that do not match to the user knowledge. Those techniques are associated with interestingness measures that may not satisfy the downward closure property, so it is impossible to integrate them in Apriori like extraction algorithms.

An element to consider is the nature of Knowledge Discovery in Databases (KDD) as an interactive and iterative user-centered process. Enforcing constraints during the mining runs neglects the character of KDD [31], [32]. A single and possibly expensive mining run is accepted but all subsequent mining questions are supposed to be satisfied with the initial result set.

In this work, a method is developed to obtain non-redundant association rules about user knowledge. It is important to ensure the user capability to refine his/her knowledge in an interactive and iterative way, accepting any of the discovered associations or discarding some previous associations and updating prior knowledge. This approach also makes possible to fulfill the mining question of different users, with different domain knowledge, in a single mining run.

## 3 A knowledge guided approach

### 3.1 Knowledge based redundancy

In example 1, a group of redundant rules, which are currently not covered by the definitions of redundancy, are showed. Our interest is to eliminate these forms of redundancy in association rule models. Based on a core set of rules that represent the user belief; a result of his experience working in the subject area. This knowledge is more general than rules obtained in the mining process which
only represent a particular dataset with partial information so the quality metric value for this kind of rule is considered maximal. This set of rules will be named prior knowledge. A rule that does not contradict prior knowledge of the user will be considered redundant. We formalize the notion of prior knowledge redundancy in definition 4. User can represent previous knowledge in different ways like semantic networks, ontologies, among others.

Considering that, the expert is interested in association rules discovering; prior knowledge is incorporated to the model using association rules format. For example an expert working with the dataset presented in table 1 knows that customers with high income ([income][high]) pay their loans on time and therefore these must be approved.

This knowledge can be represented as the association rule {
    [income][high] → {loan][yes]
}

Definition 4. Knowledge Based Redundancy: Let $S$ be a set of association rules and $S_c$ a set of prior known rules, defined over the same domain of $S$. An association rule $R: X \rightarrow Y \in S$ is redundant with respect to $S_c$ if there is a rule $R' : X' \rightarrow Y' \in S_c$ and fulfills some of the following conditions.

1. $X' \subseteq X \wedge Y' \cap Y \neq \emptyset$
   A rule is redundant if there is another rule presented in $S_c$ that contains more general information.

2. $X' \subseteq X \wedge \exists R'': X'' \rightarrow Y'' \in S_c : X'' \subseteq Y' \wedge Y \subseteq Y''$
   A rule is redundant if there is a rule $R''$ in $S_c$ that contains part or the whole antecedent and there is a third rule $R'$ in $S_c$ that shares information with $R'$ and its consequent contains $R$ consequent.

3. $X' \subseteq X \wedge Y' \cap X \neq \emptyset$
   A rule is redundant if its antecedent contains a part or the whole information of a previously known rule.

4. $X' \subseteq Y \wedge Y' \cap Y \neq \emptyset$
   A rule is redundant if its consequent contains a part or the whole information of a previously known rule.

Reviewing rules example 1 with definition 4 we have:

$S_c = \{[\{\text{income}[\text{high}]\} \rightarrow \{\text{loan}[\text{yes}]\}, [\{\text{sex}[\text{female}], \text{unemployed}, \text{yes}\}] \rightarrow \{\text{income}[\text{high}]\}\}$

Rule $R_3 : \{[\text{sex}[\text{female}], \text{unemployed}, \text{no}\} \rightarrow \{\text{income}[\text{high}], \text{loan}, \text{yes}\}$ fulfills condition 1 in definition 4 because:

1. $[\text{sex}[\text{female}], \text{unemployed}, \text{no}] \subseteq [\text{sex}[\text{female}], \text{unemployed}, \text{no}]$
2. $[\text{income}[\text{high}] \subseteq [\text{income}[\text{high}], \text{loan}, \text{yes}]$

Rule $R_4 : \{[\text{sex}[\text{female}], \text{unemployed}, \text{no}\} \rightarrow \{\text{income}[\text{high}], \text{loan}, \text{yes}\}$ fulfills condition 2 in definition 4 because:

1. $[\text{sex}[\text{female}], \text{unemployed}, \text{no}] \subseteq [\text{sex}[\text{female}], \text{unemployed}, \text{no}]$
2. $[\text{income}[\text{high}] \subseteq [\text{income}[\text{high}], \text{loan}, \text{yes}]$

Rule $R_5 : \{[\text{income}[\text{high}], \text{loan}, \text{yes}] \rightarrow \{\text{unemployed}, \text{no}\}$ fulfills condition 3 in definition 4 because:

1. $[\text{income}[\text{high}] \subseteq [\text{income}[\text{high}], \text{loan}, \text{yes}]$
2. $[\text{loan}, \text{yes}] \subseteq [\text{income}[\text{high}], \text{loan}, \text{yes}]$

Armstrong’s axioms [33] are a set of inference rules. They allow to obtain the minimum set of functional dependencies that are maintained in a database. The rest of functional dependencies can be derived from this set. They are part of clear mechanisms designed to find smaller sub-sets of a larger set of functional dependencies called “covers” that are equivalent to the “bases” in Closure Spaces and Data Mining.

Armstrong’s axioms cannot be used as an inference mechanism in association rules [34] because it is impossible to obtain the values of support and confidence in the derived rules:

- Reflexivity (if $A \subseteq A$ then $A \rightarrow B$) holds because $\text{conf}(A \rightarrow B) = \frac{\text{supp}(A \cap B)}{\text{supp}(A)} = \frac{\text{supp}(A)}{\text{supp}(A)} = 1$
- Transitivity if $A \rightarrow B$ and $B \rightarrow C$ both hold with confidence $\geq \text{threshold}$ we can not know the value for $\text{conf}(A \rightarrow C)$ so the Transitivity does not hold.
- Augmentation (if $A \rightarrow B$ then $AC \rightarrow B$) does not hold. Enlarging the antecedent of a rule may give a rule with much smaller confidence, even zero: think
of a case where most of the times \( X \) appears it comes with \( Z \), but it only comes with \( Y \) when \( Z \) is not present; then the confidence of \( X \rightarrow Z \) may be high whereas the confidence of \( XY \rightarrow Z \) may be null.

Our intention is to use Armstrong’s axioms in order to assess if a rule has Prior Knowledge Redundancy over a set of rules \( S_c \) from previous knowledge. So they must verify the condition presented in definition 4.

Condition \( X' \subseteq X \land Y' \cap Y \neq \{0\} \) represents the classical definition of redundancy like in definition 1, definition 2 and definition 3. This condition is fulfilled if a single attribute in flexitivity axiom on be rules. Suppose \( Y' \cap Y = Y_1 \). Then by the reflexivity axiom on \( R_2 \) consequent \( R_3 : Y \rightarrow Y_1 \) and by reflexivity on \( R_1 \) consequent \( R_4 : Y \rightarrow Y_1 \). By transitivity between \( R_1 \) and \( R_3 \) we have \( R_5 : X \rightarrow Y_1 \), applying transitivity between \( R_2 \) and \( R_4 \) we have \( R_6 : X \rightarrow Y_1 \). \( X \subseteq X \) by statement condition, applying augmentation in \( R_6 \) until \( X' = X, R_7 : X \rightarrow Y_1 \). Therefore Armstrong’s axioms check the condition. For example, the rule \( R : \{ [\text{income}, \text{high}] \} \rightarrow \{ [\text{loan}, \text{yes}] \} \) and prior knowledge \( S_c = \{ R_{s1} : [\text{income}, \text{high}] \rightarrow [\text{loan}, \text{yes}] \} \) is redundant according to condition 2 in definition 4. \( R \) is a single consequent rule so no separation is needed. By the application of Transitivity between \( [\text{income}, \text{high}] \rightarrow [\text{loan}, \text{yes}] \) and \( S_c \) the rule \( R_1 : [\text{sex}, \text{female}, \text{unemployed}, \text{no}] \rightarrow [\text{loan}, \text{yes}] \) is obtained \( R = R_1 \) so \( R \) is a redundant rule.

Condition \( X' \subseteq X \land Y' \cap X \neq \{0\} \) represents the case when any item in the antecedent of a rule is a redundant one. Let \( R_1 : X \rightarrow Y \) and \( R_2 : X' \rightarrow Y' \) be rules. Suppose \( Y' \cap X = X_1 \). Then by augmentation of \( X_1 \) in \( R_2 \) we have \( R_3 : X_1 \rightarrow X_1Y' \) and by transitivity between \( R_3 \) and \( R_3 \) \( R : X \rightarrow Y \). Therefore Armstrong’s axioms fulfill the condition. For example, with \( R : [\text{income}, \text{high}] \rightarrow [\text{loan}, \text{yes}] \) and \( S_c \) \( S_c = \{ R_{s1} : [\text{income}, \text{high}] \rightarrow [\text{loan}, \text{yes}] \} \) is obtained by Transitivity between \( R_1 \) and \( R \) we have \( R_2 : [\text{income}, \text{high}] \rightarrow [\text{income}, \text{high}] \) and \( R_3 \) is simpler than \( R \) with the same information so \( R \) is a redundant rule. However, by Augmentation of \([\text{loan}, \text{yes}] \) in \( R_2 \) we have \( R_4 : [\text{income}, \text{high}, \text{loan}, \text{yes}] \rightarrow [\text{unemployed}, \text{no}] \). Therefore Armstrong’s axioms fulfill the condition. For example, the rule \( R : \{ [\text{balance}, \text{high}] \} \rightarrow \{ [\text{income}, \text{high}] \} \) and prior knowledge \( S_c = \{ R_{s1} : [\text{income}, \text{high}] \rightarrow [\text{loan}, \text{yes}] \} \) is redundant according to condition 4 in definition 4. Applying Reflexivity of \([\text{income}, \text{high}] \) in \([\text{income}, \text{high}] \rightarrow [\text{loan}, \text{yes}] \) rule \( R_1 : [\text{income}, \text{high}] \rightarrow [\text{income}, \text{high}] \) is obtained by Transitivity between \( R_1 \) and \( R \) we have \( R_2 : [\text{income}, \text{high}] \rightarrow [\text{income}, \text{high}] \) and \( R_3 \) is also redundant.

Condition \( X' \subseteq Y \land Y' \cap Y \neq \{0\} \) represents the case when any item in the consequent of a rule is redundant with respect to other item in consequent. This condition is fulfilled if a single attribute in \( Y \) is redundant. Let \( R_1 : X \rightarrow Y \) and \( R_2 : X' \rightarrow Y' \) be rules. Suppose \( Y' \cap Y = Y_1 \). Then by the reflexivity axiom on \( R_2 \) consequent \( R_3 : Y \rightarrow Y_1 \) by transitivity between \( R_1 \) and \( R_4 \) we have \( R_5 : X \rightarrow Y_1 \). By statement condition \( X' \subseteq Y \) so by reflexivity on \( R_2 \) consequent we have \( R_6 : Y \rightarrow X' \). By transitivity between \( R_2 \) and \( R_6 \) we have \( R_7 : X \rightarrow X' \) now by transitivity between \( R_2 \) and \( R_7 \) we have \( R_8 : X' \rightarrow Y' \). Applying augmentation in \( R_8 \) until we have \( R_9 : X \rightarrow Y' \). By reflexivity in \( R_9 \) consequent \( R_{10} : Y \rightarrow Y_1 \) and by transitivity between \( R_9 \) and \( R_{10} \) we have \( R_{11} : X \rightarrow Y_1 \). Therefore Armstrong’s axioms check
We do not use Armstrong’s Axioms as an inference mechanism so, we do not worry if it is not able to ensure the support and confidence threshold in the inferred rules.

3.2 Algorithm to eliminate prior knowledge redundancy in association rules

In this section we present an algorithm to determine if a rule contains redundant items, see Fig. 1. The closure algorithm presented in [35] is used to compute $X^+$.

Require: Set of previous knowledge rules $S_c$
A rule $R_i$ in form $X \rightarrow Y$

Ensure: Boolean value to indicate if the rule is redundant

1: $i = 0$
2: $n = |Y|$
3: while $i < n$ do
4: if $Y[i] \in X^+_S \cup \{X - (Y[i])\}$ then
5: return true
6: end if
7: $i = i + 1$
8: end while
9: $n = |X|$ 
10: while $i < n$ do
11: if $X[i] \in (X - X[i])^+_S \cup \{X - X[i]\} \rightarrow Y$ then
12: return true
13: end if
14: $i = i + 1$
15: end while
16: return false

Algorithm 1: Prior Knowledge Redundancy detection

To determine the redundancy of a rule $X \rightarrow Y$ we have to prove if any item $A$ in the rule’s antecedent is redundant or if an item $W$ in the consequent is redundant. The item $A$ is redundant if the consequent can be derived from the prior knowledge without $A$. The first part of algorithm 1 performs this task for all items $A \in X$ by calculating the closure of the new antecedent $X - \{A\}$ over the previous knowledge rules joined to the studied rule focus, and comparing results with the closure of the same antecedent over the set of previous rules joined to a new rule, where the item $A$ is not a part of the antecedent. If both results are equal, then the item $A$ is redundant and the entire rule is also redundant. To test if item $W$ is redundant we have to apply a similar procedure, the second part of algorithm 1 performs this task.

Example 2. Prior Knowledge Redundancy detection: We use the following Prior Knowledge

$S_c = \{R_{i1} : \{income, [high]\} \rightarrow [loan], [yes]\},$

$R_{i2} : \{sex][female], [unemployed], [no] \rightarrow [income, [high]]\}$ and the rules

$R_1 : \{[balance, [high], [unemployed], [no]] \rightarrow [income, [high], [loan], [yes]]\}$

$R_2 : \{[income, [high], [loan], [yes]] \rightarrow ([unemployed], [no])\}$

The first step is to compute $F = S_c \cup R_i$ for $R_1$

$F = \{R_{i1} : [income, [high]] \rightarrow [loan], [yes], R_{i2} : [sex][female], [unemployed], [no] \rightarrow [income, [high]], R_{i3} : [balance, [high], [unemployed], [no]] \rightarrow [income, [high], [loan], [yes]]\}.$

Second, checks the redundancy in the antecedent, computing closure of $[balance, [high]]$ over $F$. This is $[balance, [high]]_F^+ = [balance, [high]]$ and comparing with closure of $[balance, [high]]$ over $G$

where $G = ((F - \{R_{i1}\}) \cup ([balance, [high]])) \rightarrow [income, [high], [loan], [yes]], [balance, [high]]_G^+ = [balance, [high], [income, [high], [loan], [yes]]\}$

They are different so $[unemployed], [no]$ is not redundant. The item $[balance, [high]]$ is also non-redundant.

And last, checks the redundancy in the consequent.

$F^\prime = \{(F - R_{i1}) \cup ([balance, [high]]), [unemployed, [no] \rightarrow [income, [high]])\} \rightarrow [balance, [high], [unemployed], [no]]_F^+ = [balance, [high], [unemployed], [no]], [income, [high], [loan], [yes]], [balance, [high], [unemployed], [no]]_F = [balance, [high], [unemployed], [no], [income, [high], [loan], [yes]]\}$

They are the same so the item $[loan], [yes]$ and the rule $R_1$ are redundant.

For $R_2$ we have:

$- F^\prime = (F - R_{i1}) \cup [income, [high]] \rightarrow [unemployed, [no]], F = ((F - R_{i1}) : [income, [high]] \rightarrow [loan], [yes], R_{i2}[sex][female], [unemployed, [no] \rightarrow [income, [high], R_{i3}[income, [high], [loan], [yes]] \rightarrow [unemployed, [no]].$

$- [income, [high]]_F^+ = [income, [high], [loan], [yes], [unemployed, [no]], [income, [high]]_F = [income, [high], [loan], [yes], [unemployed, [no]].$

They are the same so the rule is redundant.

3.2.1 Correctness

We first prove that closure algorithm [35] can be used to detect redundancy according to definition 4. Closure algorithm applies Armstrong’s axioms to find all items implied by a given itemset.

Theorem 1. Let $S_c$ be a set of prior known rules and $R : X \rightarrow Y$ an association rule. If there is a rule $R^\prime : X^\prime \rightarrow Y^\prime \in S_c$ and $X^\prime \subseteq X \land Y^\prime \cap Y \neq \emptyset$ then $Y^\prime \cap Y \in X^+_S$.

Proof. Assume $X^\prime \subseteq X \land Y^\prime \cap Y \neq \emptyset$. Then $X^\prime \in X^+_S$ by assumption $X^\prime \subseteq X$ and reflexivity axiom. So
Therefore \( Y' \in X_{S_0}' \) by transitivity between \( X \rightarrow X' \) and \( X' \rightarrow Y' \). Therefore \( Y' \cap Y \in X_{S_0}' \) by definition of set intersection.

**Theorem 2.** Let \( S_c \) be a set of prior known rules and \( R : X \rightarrow Y \) one association rule. If there is a rule \( R' : X' \rightarrow Y' \in S_c \) and \( X' \subseteq X \wedge \exists R'' : X'' \rightarrow Y'' \in S_c : X'' \subseteq Y' \wedge Y' \subseteq Y'' \) then \( Y \in X_{S_0}' \).

**Proof.** Assume \( X' \subseteq X \wedge \exists R' : X' \rightarrow Y' \in S_c \) and \( X' \subseteq X \wedge Y' \subseteq Y'' \). Then \( X' \in X_{S_0}' \) by assumption \( X' \subseteq X \) and reflexivity axiom. \( Y' \in X_{S_0}' \) by transitivity between \( X \rightarrow X' \) and \( X' \rightarrow Y' \). \( X' \in X_{S_0}' \) by assumption \( X' \subseteq Y' \) and subset definition. So \( Y'' \in X_{S_0}' \) by transitivity between \( X \rightarrow X' \) and \( X' \rightarrow Y'' \). Therefore \( Y \in X_{S_0}' \) by assumption \( Y \subseteq Y'' \) and subset definition.

**Theorem 3.** Let \( S_c \) be a set of prior known rules and \( R : X \rightarrow Y \) one association rule. If there is a rule \( R' : X' \rightarrow Y' \in S_c \) and \( X' \subseteq X \wedge Y' \cap Y \neq \emptyset \) then \( Y' \cap Y \in (X - (Y \cap X))_{S_0}' \).

**Proof.** Assume \( X' \subseteq X \wedge Y' \cap Y \neq \emptyset \). Then \( X' \in (X - (Y' \cap X))_{S_0}' \) by assumption \( X' \subseteq X \) and reflexivity axiom. \( Y' \in (X - (Y \cap X))_{S_0} \) by transitivity between \( X \rightarrow X' \) and \( X' \rightarrow Y' \). Therefore \( Y' \cap Y \in (X - (Y' \cap X))_{S_0} \) by definition of set intersection.

**Theorem 4.** Let \( S_c \) be a set of prior known rules and \( R : X \rightarrow Y \) one association rule. If there is a rule \( R' : X' \rightarrow Y' \in S_c \) and \( X' \subseteq X \wedge Y' \cap Y \neq \emptyset \) then \( Y' \cap Y \in X_{S_0} \cap X - (Y - (Y' \cap Y)) \).

**Proof.** Assume \( X' \subseteq X \wedge Y' \cap Y \neq \emptyset \). Then \( X' \in X_{S_0} \cap X - (Y - (Y' \cap Y)) \) by assumption \( X' \subseteq X \) and association rule property \( X \cap Y = \emptyset \). \( Y' \in X_{S_0} \cap X - (Y - (Y' \cap Y)) \) by transitivity between \( X \rightarrow X' \) and \( X' \rightarrow Y' \). Therefore \( Y' \cap Y \in X_{S_0} \cap X - (Y - (Y' \cap Y)) \) by definition of set intersection.

**Theorem 5.** If \( (\exists A_i \in X \wedge A_i \in (X - A_i)_{S_0} \cup (X - A_i) \cap Y) \Rightarrow (\exists W_i \in Y \wedge W_i \in X_{S_0} \cup X - Y - W_j) \) then rule \( X \rightarrow Y \) has prior knowledge redundancy over \( S_c \).

**Proof.** Direct from theorem 1, theorem 2, theorem 3 and theorem 4.

Hoare triple was introduced by C. A. R. Hoare [38] as \{P\}C{Q}, for specifying what a program does. In such a Hoare triple:

- \( C \) is a program.
- \( P \) and \( Q \) are assertions, conditions on the program variables used in \( C \). They will be written using standard mathematical notation together with logical operators. We can use functions and predicates to express high-level properties based on a domain theory [39] covering specifics of the application area.

We say \{P\}C{Q} is true, if whenever \( C \) is executed in a state satisfying \{P\} and the execution of \( C \) finishes, then the state in which \( C \) execution finishes satisfies \( Q \). If there is a loop in \( C \), loop invariants must be used to prove correctness. If loop invariants are proved to be true after each loop iteration then the postcondition must be proven true.

In algorithm 1 lines one through eight and lines nine through sixteen perform basically the same operation, one over the rule antecedent and the other over the rule consequent. So we analyze them only one time. Line four checks if \( Y[i] \) is subset of the closure. So closure algorithm must be computed, this algorithm has been proved as correct [35]. The search of \( Y[i] \) within closure can be done by a well known linear search algorithm, we assume it is correct.

**Preconditions:**
- \( S_c \) is a set of previous knowledge rules.
- \( X \rightarrow Y \) is an association rule with \( X = X_1, \ldots X_n \) and \( Y = Y_1, \ldots Y_m \)

**Postcondition:** If \( (\exists A_i \in X \wedge A_i \in (X - A_i)_{S_0} \cup (X - A_i) \cap Y) \Rightarrow (\exists W_i \in Y \wedge W_i \in X_{S_0} \cup X - Y - W_j) \) the return value is \( true \).

**Loop invariants:** If the loop is executed \( j \) times or more, then after \( j \) executions

- \( i = j \)
- \( 0 \leq i \leq n \)
- \( Y[h] \notin X_{S_0} \cap X - (Y - (Y[i])) \) for \( 0 \leq h < i \)

**Proving the loop invariant:** (by induction on \( j \))

**Base Case:** \( j = 0 \)

- before first execution of loop \( i = 0 \)
- loop invariant holds, \( i = 0 \Rightarrow (0 \leq h < 0) \). No such \( h \) value.

**Inductive hypothesis:** assume that, if the loop iterates \( j \) times then the loop invariant holds \( i_{old} = j \). Proving that if the loop iterates \( j + 1 \) times, then the loop invariant holds for \( i_{new} = j + 1 \). If true for iteration \( i_{old} = j \) then \( Y[h] \notin X_{S_0} \cap X - (Y - (Y[i_{old}])) \) for \( 0 \leq h < i_{new} \).

- if loop iterates then \( Y[i_{old}] \notin X_{S_0} \cap X - (Y - (Y[i_{old}])) \) and \( i_{new} = i_{old} + 1 \).
- thus \( Y[h] \notin X_{S_0} \cap X - (Y - Y[h]) \) for \( 0 \leq h < i_{new} \).
- because loop iterated for \( i_{old} = j \) we have \( i_{old} < n \) and \( i_{new} \leq n \).
Thus, the loop invariant holds for \( j + 1 \).

When the loop test fails, the loop invariant holds and either \( i \geq n \) or \( Y[i] \in X_{S \cup X \rightarrow (Y \setminus Y[i])}^+ \)

- **Case 1** \( (j \geq n) \): loop invariant implies that \( Y[h] \notin X_{S \cup X \rightarrow (Y \setminus Y[h])}^+\) for \( 0 \leq h < n \), so no element in consequent is a redundant one.

- **Case 2** \( (j < n) \): loop invariant implies that \( Y[i] \in X_{S \cup X \rightarrow (Y \setminus Y[i])}^+ \) and \textit{true} is returned

\textbf{Conclusions:} Poscondition is satisfied in either case, so the algorithm is correct.

### 3.2.2 Complexity analysis

Time complexity of an algorithm is a function \( T(n) \) limiting the maximum number of steps in the algorithm for an input size \( n \). \( T(n) \) depends on what is counted as one computation step, the random access machine (RAM) model is the most extended one. RAM is a model for a simple digital computer with random access memory. For the sake of simplicity \( T(n) \) is approximated by a simplest function, it is written \( T(n) = O(f(n)) \) if there are constants \( c \geq 0 \) and \( n_1 \geq 0 \) such that: \( T(n) \leq cf(n) \) for all \( n \geq n_1 \).

For algorithm in Fig 1 we considered \( a \) as the number of different attribute symbols in \( S_c \) and \( p \) the number of previous knowledge rules presented in \( S_c \). The complexity order to compute the closure is \( O(n) \) see [35]. The execution time of the first \textbf{while} loop (the consequent of the rule) takes \( a \times p \) since the number of rules in \( F \) is \( p \), and we compute the closure with a cost of \( O(p) \). The execution time of the second \textbf{while} loop (the antecedent of the rule) takes the same value of \( a \times p \) because it performs the same operation and in the same way the complexity of the steps is \( O(ap) \). To compute the complexity of the entire algorithm, the complexity of the first and second \textbf{while} loops must be added so it is \( O(ap) + O(ap) = 2O(ap) \) but the constant 2 can be ignored and the final value for complexity of the algorithm is \( O(ap) \).

Association rules extraction algorithms have much higher complexity [36] than the reduction approach presented here. This difference led us to propose a reduction mechanism in which rule extraction algorithm is executed once and then, in the post processing stage, the reduction algorithm is fired to prune the redundant rules, rather than applying prior knowledge as restriction within the extraction algorithms, which would force to execute it for each different user and even for each change on a user’s prior knowledge. The computational cost for the constraint approach is very high. However, our approach, in post processing stage, allows us to run a simpler routine when the user changes or the user prior knowledge is updated. The temporal cost of this approach did not exceed 5 seconds in any of the applied tests.

### 4 Experimental results

#### 4.1 Methodology

In order to verify the effectiveness of our approach we performed experiments with four datasets. The first one with data about USA census[2], the second one with data about stock market investments [1], the third one with data about hypothetical samples of mushroom[2] and the last one with data about breast cancer[2]. Prior knowledge consists of 6 rules for each dataset. We use Pruning Ratio metric \( PR = \frac{\text{Pruned Rules}}{\text{Total Rules}} \times 100 \) to evaluate our results.

Table 2 shows the result of the experiments. Each row corresponds to an experiment following the next steps:

1. Find the complete set of rules using as support threshold the value in column 2 and confidence threshold the value in column 3. The number of rules is shown in column 4.

2. Apply the steps presented in algorithm 1. The number of pruned rules are presented in column 5 of Table 2.

3. After applying the algorithm to the dataset, the final number of rules is presented in column 6 of Table 2 while column 7 contains the pruning ratio. The execution time is presented in column 8.

#### 4.2 Results and discussion

Pruning Ratio changes according to support in Census and Stocks datasets, first increasing while the support increases, but when the support is greater than 0.07 for the Census dataset and greater than 0.5 for Stocks dataset, the Pruning Ratio decreases while the support increases. The behavior in Mushroom dataset is the opposite, the Pruning Ratio decreases while support increases until the support reaches the 0.5 value then the Pruning Ratio increases while the support value increases.

This behavior shows a relation between support and previous knowledge patterns. If the support is increased, then a number of rules do not meet the support threshold and they are discarded. Hence the discarded rules have no major impact on the rules derived from previous knowledge. Pruning Ratio will be increased, but as the support increases it starts to reduce the rules derived from previous knowledge, so the Pruning Ratio will be decreased.

In Fig 1, Fig 2 and Fig 3 the mean value of Pruning Ratio is shown for several support values in Census, Stocks and Mush datasets respectively using combination of all six rules in \( S_c \).
### Dataset Support Confidence Rules Pruned Rules Final Rules Pruning Ratio Time

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Support</th>
<th>Confidence</th>
<th>Rules</th>
<th>942</th>
<th>2466</th>
<th>27</th>
<th>0.589</th>
</tr>
</thead>
<tbody>
<tr>
<td>Census</td>
<td>0.01</td>
<td>0.4</td>
<td>3408</td>
<td>242</td>
<td>593</td>
<td>28</td>
<td>0.079</td>
</tr>
<tr>
<td>Census</td>
<td>0.03</td>
<td>0.4</td>
<td>835</td>
<td>158</td>
<td>300</td>
<td>32</td>
<td>0.043</td>
</tr>
<tr>
<td>Census</td>
<td>0.07</td>
<td>0.4</td>
<td>458</td>
<td>158</td>
<td>300</td>
<td>32</td>
<td>0.043</td>
</tr>
<tr>
<td>Census</td>
<td>0.09</td>
<td>0.4</td>
<td>229</td>
<td>79</td>
<td>150</td>
<td>34</td>
<td>0.021</td>
</tr>
<tr>
<td>Census</td>
<td>0.11</td>
<td>0.4</td>
<td>163</td>
<td>79</td>
<td>112</td>
<td>31</td>
<td>0.015</td>
</tr>
<tr>
<td>Stocks</td>
<td>0.2</td>
<td>0.4</td>
<td>11010</td>
<td>5592</td>
<td>5418</td>
<td>50</td>
<td>2.170</td>
</tr>
<tr>
<td>Stocks</td>
<td>0.3</td>
<td>0.4</td>
<td>3314</td>
<td>2225</td>
<td>1089</td>
<td>67</td>
<td>0.536</td>
</tr>
<tr>
<td>Stocks</td>
<td>0.4</td>
<td>0.4</td>
<td>1230</td>
<td>904</td>
<td>326</td>
<td>73</td>
<td>0.116</td>
</tr>
<tr>
<td>Stocks</td>
<td>0.5</td>
<td>0.4</td>
<td>349</td>
<td>294</td>
<td>55</td>
<td>84</td>
<td>0.039</td>
</tr>
<tr>
<td>Stocks</td>
<td>0.6</td>
<td>0.4</td>
<td>212</td>
<td>64</td>
<td>148</td>
<td>30</td>
<td>0.020</td>
</tr>
<tr>
<td>Mushroom</td>
<td>0.3</td>
<td>0.5</td>
<td>78998</td>
<td>29154</td>
<td>49844</td>
<td>36</td>
<td>11.245</td>
</tr>
<tr>
<td>Mushroom</td>
<td>0.4</td>
<td>0.5</td>
<td>5767</td>
<td>1225</td>
<td>4542</td>
<td>21</td>
<td>0.852</td>
</tr>
<tr>
<td>Mushroom</td>
<td>0.5</td>
<td>0.5</td>
<td>1148</td>
<td>200</td>
<td>948</td>
<td>17</td>
<td>0.098</td>
</tr>
<tr>
<td>Mushroom</td>
<td>0.6</td>
<td>0.5</td>
<td>266</td>
<td>88</td>
<td>178</td>
<td>33</td>
<td>0.025</td>
</tr>
<tr>
<td>Mushroom</td>
<td>0.7</td>
<td>0.5</td>
<td>180</td>
<td>83</td>
<td>97</td>
<td>46</td>
<td>0.017</td>
</tr>
<tr>
<td>Breast</td>
<td>0.01</td>
<td>0.4</td>
<td>210500</td>
<td>98582</td>
<td>111918</td>
<td>47</td>
<td>27.732</td>
</tr>
<tr>
<td>Breast</td>
<td>0.1</td>
<td>0.4</td>
<td>28808</td>
<td>13695</td>
<td>15113</td>
<td>47</td>
<td>4.190</td>
</tr>
<tr>
<td>Breast</td>
<td>0.2</td>
<td>0.4</td>
<td>6092</td>
<td>2982</td>
<td>3110</td>
<td>49</td>
<td>0.859</td>
</tr>
<tr>
<td>Breast</td>
<td>0.3</td>
<td>0.4</td>
<td>5284</td>
<td>2398</td>
<td>2886</td>
<td>45</td>
<td>0.798</td>
</tr>
<tr>
<td>Breast</td>
<td>0.4</td>
<td>0.4</td>
<td>1246</td>
<td>449</td>
<td>797</td>
<td>36</td>
<td>0.118</td>
</tr>
</tbody>
</table>

Table 2: Experiment’s result

### 4.3 Traditional vs. knowledge based reduction

The approach developed in this paper differs from those published until now. Previous works are concerned with the structural relationship between association rules and mechanisms to reduce redundancy using inference rules and maximal itemsets. We use the user experience to prune rules that do not bring new knowledge to the user, simplifying decision making. Both approaches are not comparable in essence, but we carried out experiments to compare KBR’s pruning ratio with previous works.

Fig 4 shows the pruning ratio of some relevant works in redundancy reduction, over a Mushroom dataset with a support value of 0.3. We used Mushroom dataset because we can access to author experiments and it is sufficient to test our case. The values for pruning ratio are taken from the author’s papers: MinMax, Reliable, GB, CHARM, CRS and MetaRules.[40]

Reliable has the best Pruning Ratio, see Fig 4, so we compare it with our approach at different support values, see Table 3.

With very few rules in KBR is possible to exceed the Pruning Ratio of previous works. Of course there is a narrow relationship between the Pruning Ratio and the repercussion of the previous knowledge rules over the whole set of rules. The Pruning Ratio of knowledge rules increases in the same way that they are able to describe the domain under study. The better KBR results are, the better the user will know the domain under study. Our approach has the possibility to determine when a model can not be improved like in the case of $KBR_{15 rules}$ for a support value of 0.7 where the Pruning Ratio is 100%.

### 4.4 Knowledge vs knowledge based reduction

In section 2 we surveyed some works that used knowledge to reduce the number of association rules presented to the final user. The main goal of those papers is to obtain a set of association rules that satisfies some constraint provided by users, using different forms of knowledge representation. They are able to reduce the association rules set cardinality but generate a lot of rules that represent the same knowledge. Strictly speaking we can not compare our proposal with those ones because of the difference between goals, but we want to test the association rules model cardinality reduction capability of our approach with template, the best known form of knowledge approach.

We compare the pruning ratio of our approach with the template implementation proposed in [41] that up-perform the implementation proposed in [16] across five dataset from [2].
Knowledge Redundancy Approach to Reduce Size in...

Informatica 44 (2020) 167–181

Figure 1: Rules pruned in census dataset

Table 3: Pruning Ratio

<table>
<thead>
<tr>
<th>Support</th>
<th>Reliable</th>
<th>$KBR_{0.01rules}$</th>
<th>$KBR_{0.03rules}$</th>
<th>$KBR_{0.05rules}$</th>
<th>$KBR_{0.07rules}$</th>
<th>$KBR_{0.09rules}$</th>
<th>$KBR_{0.11rules}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>95</td>
<td>36</td>
<td>76</td>
<td>80</td>
<td>96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>90</td>
<td>21</td>
<td>37</td>
<td>47</td>
<td>84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>89</td>
<td>17</td>
<td>30</td>
<td>44</td>
<td>93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>74</td>
<td>33</td>
<td>40</td>
<td>62</td>
<td>97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>78</td>
<td>46</td>
<td>46</td>
<td>75</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>85</td>
<td>32.5</td>
<td>45.8</td>
<td>61.5</td>
<td>94</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Remaining rules

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TotalRules</th>
<th>MetaRules</th>
<th>KBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>mush</td>
<td>1374</td>
<td>138</td>
<td>120.2</td>
</tr>
<tr>
<td>ion</td>
<td>1215</td>
<td>452</td>
<td>402.6</td>
</tr>
<tr>
<td>hea</td>
<td>371</td>
<td>246</td>
<td>176.7</td>
</tr>
<tr>
<td>thy</td>
<td>1442</td>
<td>502</td>
<td>431.6</td>
</tr>
<tr>
<td>smo</td>
<td>797</td>
<td>300</td>
<td>283.3</td>
</tr>
</tbody>
</table>

5 Conclusion

The continuous attributes in the data sets used were discretized using a 4-bin equal-frequency discretization. Support and Confidence were set to the same values used in [16]. In table 4 we present the result of our pruning approach (KBR) and compare it with the previous work (MetaRules) [41].

Each row in table 4 represents an experiment where column Dataset contains the dataset id, column TotalRules shows the total number of rules produced by extraction algorithms, MetaRules presents the remaining rules after the application of the algorithm proposed in [41] while column KBR contains the average of remaining rules of ten runs of knowledge based redundancy elimination algorithm using a random knowledge of ten rules for each execution. The remaining rules in our approach are lower than the number of rules in metarules approach for all datasets.

The fundamental idea in this work is linked to the main definition of data mining: analysis of large amount of data to extract interesting patterns, previously unknown and the consideration that an association rule that correspond to prior knowledge is a redundant one[37]. Our approach prunes those rules, presenting a simpler model to the final user.

The main contribution in this work is the definition of redundancy of association rules with respect to prior knowledge, and the definition of a mechanism to eliminate this kind of redundancy from the final model of association.
rules presented to the end user. The redundancy elimination is performed in two procedures, the first one to detect and prune redundant element in rules antecedent and consequent, and the second one to detect if all information provided by a rule is redundant with respect to prior knowledge and then to prune it.

The results of this study confirm it is possible to use prior knowledge of experts to reduce the volume of association rules. Models of association rules with fewer rules can be interpreted more clearly by specialists so they can generate advantages in decision making process. The experimental results show that prior knowledge of less than 10% can reach a reduction ratio above 90%.

Acknowledgement

This research is partially funded by the research project TIC1582 by “Consejería de Economía, Innovacion, Ciencia y Empleo from Junta de Andalucía” (Spain).

References


Figure 3: Rules pruned in Mushroom dataset


Figure 4: Pruning Ratio for different approaches.

---


what we want?: an alternative approach for association rule mining”, ACM SIGKDD Explorations 4(1), pp.50-55. https://doi.org/10.1145/568574.568582


